

# Lokalizacija

Iako više od polovine internet korisnika danas u svetu zna engleski, ovaj jezik nije jedini jezik koji koristi veliki broj ljudi. Recimo, po statistici [sajta w3tech](#), manje od 5% engleskih sajtova ima sadržaj prilagođen španskom govornom području, iako je taj jezik u svetu veoma rasprostranjen. Ukoliko bi web sajtovi bili samo na engleskom, veliki broj korisnika koji ne zna engleski ne bi imao dovoljan pristup željenom sadržaju. Kako bi se izbegao gubitak potencijalnih korisnika i zarade, često je poželjno prevesti sajt na druge jezike.

Dakle, često postoji potreba za lokalizacijom sajtova i prilagođavanjem sadržaja za određena područja. Treba imati na umu da pored ovog termina – lokalizacija, postoji i internacionalizacija, koja se odnosi na to da se olakša lokalizacija samog sadržaja.

Lokalizacija se ne odnosi samo na prevođenje sadržaja već i na adaptiranje tog sadržaja ciljanom regionu/kulturi itd. Ovo znači da se lokalizacija odnosi na valutu, jedinice mere i generalnu terminologiju specifičnu za region za koji lokalizujemo sadržaj. Zato je prethodno važno istražiti tržište za koje planiramo lokalizovan sadržaj.

## Prednosti lokalizovanja sadržaja

Prednosti lokalizovanja sadržaja su:

- ciljanje šireg auditorijuma – lokalizacija otvara vrata za nove korisnike koji, da je sadržaj na jeziku koji ne razumeju, ne bi ni posetili sajt;
- bolje rangiranje našeg sajta među rezultatima pretrage – može se kreirati poseban link ka našem sajtu koji se odnosi upravo na lokalizovanu verziju sajta, što može poboljšati vidljivost našeg sajta prilikom Google pretrage;
- sajt postaje pristupačniji – lokalizacijom ćemo olakšati pristup ljudima koji ne poznaju dobro primarni jezik našeg sajta.

O tome za koji region će se sadržaj prilagoditi se najbolje odlučuje na osnovu same analitike sajta i analize saobraćaja na sajtu.

### Pitanje

Da li ćemo lokalizovati sadržaj ako nam većina korisnika dolazi samo sa našeg govornog područja?

- Da
- **Ne**

### Objašnjenje:

*Tačan odgovor je da u slučaju kada većina posetilaca sajta dolazi sa našeg govornog područja nije potrebno lokalizovati sadržaj sajta.*

## Lokalizacija na serverskoj strani

Da li je sadržaj lokalizovan ili ne se najbolje vidi iz pretraživača, ali ćemo za naš primer koristiti Python klijent i server. Za kod servera ćemo iskoristiti sledeći kod iz kursa *Python Net Programming*:

### Serverski kod (`http_local_server_localisation.py`):

```
from http.server import HTTPServer, BaseHTTPRequestHandler
localisation_dict = {'en': 'Greetings!', 'es': 'Saludos!'}
class RequestHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        self.log_message("Incoming GET request with headers
{}".format(self.headers.items()))
        try:
            self.send_response_to_client(200,
localisation_dict[self.headers.get('Accept-Language')])
        except:
            self.send_response_to_client(404, 'Incorrect language code
provided')
            self.log_message("Incorrect language code provided")
    def send_response_to_client(self, status_code, data):
        # Send OK status
        self.send_response(status_code)
        # Send headers
        self.send_header('Content-type', 'text/plain')
        self.end_headers()
        # Send the response
        self.wfile.write(str(data).encode())
server_address = ('127.0.0.1', 8000)
http_server = HTTPServer(server_address, RequestHandler)
http_server.serve_forever()
```

Nama najbitnija linija u kodu je `self.send_response_to_client(200, localisation_dict[self.headers.get('Accept-Language')])`, kojom klijentu šaljemo statusnu poruku 200 zajedno sa porukom. Poruka se u ovom slučaju generiše pomoću rečnika `localisation_dict`, čiji su ključevi imena jezika, a vrednosti tih ključeva reč *Pozdrav* prevedena na odgovarajuće jezike (u našem primeru su to `us` – United States i `es` – España). Informaciju o jeziku dobavljamo iz klijentskog zaglavlja metodom `self.headers.get()`. Metodi `get()` nad zaglavljem našeg serverskog objekta prosleđujemo ime zaglavlja koje želimo da dobavimo iz zahteva. U slučaju da takvo polje zaglavlja ne postoji, ova metoda vraća `None`. Ako bismo želeli da dobavimo sve ključ-vrednost parove zaglavlja, umesto `get()` metode koristili bismo metodu `items()` koja vraća listu n-torki gde svaka ta n-torka sadrži samo dva elementa – ključ i vrednost (ime zaglavlja i njegova vrednost). Sama linija `self.headers` nam vraća objekat tipa `MessageClass`, čiji se atributi i metode mogu naći u [zvaničnoj dokumentaciji](#). Ako prosleđena oznaka jezika ne postoji u našem `localisation_dict` rečniku kao ključ, nastaje `KeyError` greška, ali će biti uhvaćena sa `try-except` blokom koda koji će u tom slučaju klijentu poslati poruku koja glasi „Incorrect language code provided” sa kodom 404. Polje zaglavlja u kojem tražimo oznaku jezika je `Accept-Language`.

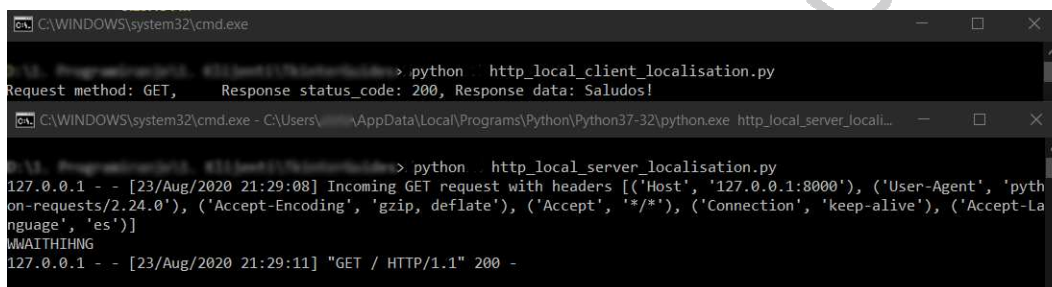
Na klijentskoj strani koristićemo biblioteku `requests` (`requests` nije deo biblioteka koje dolaze uz instalaciju Pythona, pa ju je potrebno naknadno instalirati), a kod koji koristimo je takođe sličan HTTP klijentskom kodu iz kursa *Python Net Programming* i on izgleda ovako:

### Klijentski kod (http\_local\_client\_localisation.py):

```
import requests
headers = {'Accept-Language': 'es'}
r = requests.get("http://127.0.0.1:8000/", headers = headers)
print("Request method: GET, \
      Response status_code: {}, Response data: {}".format(r.status_code,
r.text))
```

U ovom primeru smo zaglavlje prethodno definisali kao rečnik koji se prosleđuje requests.get() metodi koji sadrži upravo Accept-Language polje kojim ćemo poslati zahtev serveru da želimo sadržaj na španskom (es – España, kod za španski jezik).

Prvo treba pokrenuti serverski kod u posebnom prozoru komandne linije, a potom pokrenuti i klijentski kod. Nakon pokretanja ispis izgleda ovako:



The screenshot shows two overlapping command prompt windows. The top window, titled 'C:\WINDOWS\system32\cmd.exe', shows the execution of 'python http\_local\_client\_localisation.py', which outputs 'Request method: GET, Response status\_code: 200, Response data: Saludos!'. The bottom window, titled 'C:\WINDOWS\system32\cmd.exe - C:\Users\...\AppData\Local\Programs\Python\Python37-32\python.exe http\_local\_server\_locali...', shows the execution of 'python http\_local\_server\_localisation.py'. It displays an incoming GET request with headers including 'Accept-Language: es' and a successful response '200'.

Slika 6.1. Konzolni ispis nakon pokretanja klijentskog i serverskog koda

Ako bismo umesto es prosledili kod za francuski jezik – fr, u serverskom odgovoru bismo dobili poruku „Incorrect language code provided”, što znači da sadržaj koji odgovara tom kodu ne postoji na serveru. Naravno, ovo je loša praksa – da klijent dobija grešku ako ne može da pristupi lokalizovanom sadržaju, pa je u tom slučaju dobro da se klijentu ipak pošalje nelokalizovana, izvorna verzija sadržaja koja već postoji na serveru.

## Rezime

- Lokalizacija se ne odnosi samo na prevođenje sadržaja, već i na adaptiranje tog sadržaja za ciljani region/kulturu.
- Internacionalizacija se odnosi na olakšavanje lokalizacije samog sadržaja.
- Pre lokalizacije je važno istražiti tržište za koje planiramo da lokalizujemo sadržaj.
- Prednosti lokalizovanja sadržaja su:
  - ciljanje šireg auditorijuma – lokalizacija otvara vrata za nove korisnike koji, da je sadržaj na jeziku koji ne razumeju, ne bi ni posetili sajt;
  - bolje rangiranje našeg sajta među rezultatima pretrage – može se kreirati poseban link ka našem sajtu, koji se odnosi upravo na lokalizovanu verziju sajta, što može poboljšati vidljivost našeg sajta prilikom Google pretrage;
  - čini sajt pristupačnijim – lokalizacijom ćemo olakšati pristup ljudima koji ne poznaju dobro primarni jezik našeg sajta.